FORMAL METHOD FOR IDENTIFYING TWO TYPES OF AUTOMATION SURPRISES

Lance Sherry
Honeywell
Lance.sherry@honeywell.com

Michael Feary
SJSU/NASA
mfeary@mail.arc.nasa.gov

Peter Polson
University of Colorado
ppolson@psych.colorado.edu

Everett Palmer
NASA
epalmer@mail.arc.nasa.gov

ABSTRACT:

"Automation surprises" occur when operators of sophisticated automation, such as pilots of aircraft, hold a mental model of the behavior of the automation that does not reflect the actual behavior of the automation. This leads to increased workload, and reduced efficiency and safety.

This paper describes a formal method for analysis of automation and it's user-interface for two well known characteristics that lead to automation surprises: (1) an automation user input device that, when selected, results in different automation behaviors depending on the situation, and (2) automation displays that do not provide unique annunciation for all automation behaviors. This method is unique in that it is based on analysis of the goals and behavior of the actual automation software. This provides a meaningful basis to perform user-oriented task analyses. A case study is also provided.

Keywords:
Automation surprise, formal methods, human factors, certification criteria.

INTRODUCTION

Although automation surprises have not been cited as *the* contributing factor in any incidents or accidents, there is a consensus among researchers in aviation, that the gap between a pilot's understanding of the avionics behavior, and the actual behavior of the avionics has lead to increased workload in the cockpit (FAA, 1996; BASI, 1999). In fact many, airlines, rather than face the task of training the pilots on the operation of functions perceived to be too complex, have explicitly placarded the function, or provided training on only limited use of the automation (Hutchins, 1994). Furthermore, pilots simply choose not to use parts of the automation (Sarter, Woods, & Billings, 1997).

The root cause of automation surprises is the failure of the operator-automation system to establish a shared understanding of the situation, and agreement on the correct response to the situation (Norman, 1988; Reason, 1987; Sherry & Polson, 1999). Several characteristics of automation and their user-interfaces that lead to automation surprises have been identified. This paper describes a formal method for identifying two characteristics of the automation that lead to automation surprises:

(1) Displays that do not provide unique annunciation of all automation behavior.

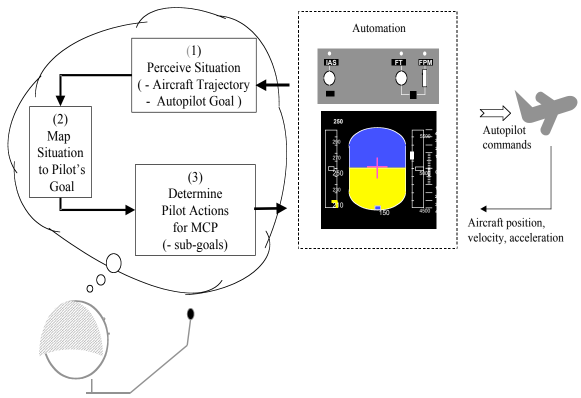
(2) Input devices that invoke more than one automation behavior depending on the situation.

This method of analysis is unique as these results are derived from analysis of the goals and behavior of the actual automation software, not hypothesized models of operator behavior. The results are obtained by analysis (no simulation is required). Furthermore, this model serves as the basis for performing other types of task analysis and user-interface evaluations.

Automation surprises and their root causes are explained in the next section. This is followed by a definition of the Situation-Goal-Action (SGA) model and a description of the method for analyzing the SGA model for the two characteristics that lead to automation surprises. A case study of this type of analysis on the NASA Research Autopilot is provided. The paper concludes with a comparison to other user-interface analysis methods and a discussion on techniques to address these automation surprise characteristics when they are present in automation.

AUTOMATION SURPRISES

The root cause of the automation surprises is mismatch between the operator's mental model of the behavior of the automation and the actual behavior of the automation. Norman (1988) proposed that operators of automated systems form "mental models" of the way the system behaves and use these models to guide their interaction with the system. This interaction with the automation (and much other human behavior) can be thought of as a continuous process of cyclic interaction (Monk, 1999; Card, Moran, & Newell, 1983; Norman, 1988; Anderson, 1993). This process is abstracted in the picto-gram in Figure 1 illustrating a pilot's interaction with the cockpit



(1) Situation, (2) goal, and (3) action sequence of pilot interaction with automation. Boxes represent knowledge required by the pilot (Sherry et.al. , submitted)
Figure 1

automation (Sherry et. al., submitted-a). Based on information from the environment, the pilot formulates a definition of the perceived situation (block 1). This situation is used to determine appropriate goals (block 2). The goals are mapped to a sequence of pilot actions on the MCP (block 3).

The ability of the operator to perform these tasks requires knowledge stored in the operator's head. The amount of knowledge, the structure of the knowledge and its retrieval cues are directly related to the form and content of the user-interface. User-interfaces organized based on the task to be performed, with adequate display of the situation, and clearly labeled user input devices, minimize the amount of knowledge that must be trained and stored in the operators head. Minimizing this knowledge provides for a more robust system, as knowledge in the operators head is subject to corruption by natural cognitive processes of infrequent use and generalization (Javaux, 1998) that result in operator mistakes (Reason, 1987) and action slips (Norman, 1988).

Example Automation Surprises

Several examples of automation surprises in aviation have been analyzed in the literature (Palmer, 1995; Degani & Heymann, in press; Javaux, 1998; NTSB, 1999). The following scenario from Sherry et. al (submitted –a) illustrates the two types of automation surprises discussed in this paper.

> **Representative example:** "When flight xxx was cleared to descend to 20,000 ft, the first officer initiated a descent via the autopilot. With approximately 1,200 ft left in the descent, *the captain became concerned the airplane might not level off at the assigned altitude* and instructed the first officer to slow the descent rate. The first officer adjusted the MCP vertical speed wheel several times; however this maneuver proved ineffective. The captain then took manual control of the airplane, and disconnected the autopilot …"

(a) "…*the captain became concerned the airplane might not level off at the assigned altitude*," describes a phenomenon in which operators question what the system is doing, and more importantly, what it is going to do next. In this case the pilot/automation system fails to establish *a shared understanding of the intention of the automation.*

(b) "…the captain …instructed the first officer to slow the rate of descent. The first officer adjusted the MCP vertical speed wheel several times; however this maneuver proved ineffective," describes a phenomenon in which the operators were unable to convey their trajectory goals to the automation. In this case the interface between the pilot and automation failed to provide *adequate affordances for the pilot to convey their goals to the autopilot.*

Sherry, et. al. (submitted –a) demonstrated that the pilots confusion about the intended behavior of the autopilot in (a) above can be attributed to automation displays that fail to explicitly identify the capture trajectory to the MCP altitude and the sequence of pitch and thrust modes to perform this maneuver. Inadequate feedback by the automation to the pilot due to non-unique annunciation of automation behavior is a major factor in the operator-automation interaction problems known as "mistakes" (Reason, 1987). Failure to train the pilot to recognize cues that lead to the correct situation awareness, and natural cognitive processes that corrupt pilot's mental models also contribute to these mistakes.

Sherry, et. al. (submitted –b) demonstrated that the pilot failed to convey the pilot goal to the autopilot in (b) above, due to the fact that the automation user input device (MCP vertical speed wheel) results in two different behaviors depending on whether the wheel is selected during a capture or not. In one case selection of the input device results a climb/descent and maintain the MCP altitude. In the other case, selection of the input device results in a climb/descend *away* from the MCP altitude. Context dependent, or modal, input devices are a major factor in the operator-automation interaction problems known as "action slips" (Norman, 1988). Failure to train the pilot to recognize the "mode" of the input device, and natural cognitive processes that corrupt the pilot's mental model also contribute to these action slips.

The SGA model, and the method for analysis of the model for two characteristics that lead to these automation surprises, is described in the next section.

METHOD OF ANALAYIS: SGA MODEL

The Situation-Goal-Action (SGA) model, a variation of the Operational Procedure Model (Sherry, 1995), layers a semantic goal-based model over a formal situation-action model of an extended finite state machine.

Situation	$= f$ (state of env. from system inputs)	(a)
Goal	$= f$ (situation)	(b)
Outputs	$= f$ (goal, actions)	(c)

The patterns of values of the inputs to the automation determine the situation perceived by the automation (equation a). The situation is used to determine the goal of the automation (equation b). The goal represents a semantic description of the behavior of the system and used is by the analyst to concisely describe the behavior. Based on the goal, a prescribed set of actions (or functions) are executed to generate values for the outputs (equation c).

The SGA model is constructed from either an input-output analysis of the software, or derived from a flow-of-control analysis of the software algorithms. The model is an aggregated rule-based model with the same behavior as the actual software. The separation of the decision-making for determining the situations, from the algebraic computation/data manipulation of the actions, provides the mechanism to perform the analysis described below. Also the assignment of situation-action pairs with goal labels provides a way to manage the complexity of the model. The goal definition is also very useful for performing user-oriented task-analysis (see Conclusions).

By definition, the SGA model defines all the legal combinations of actions for each output. This definition represents the complete set of behaviors, or goals, of the system. For example the overall behavior of a modern autopilot is defined by the values/actions on the outputs: altitude target, speed target, vertical speed target, pitch control mode, and thrust control mode. Different combinations of values/actions on the outputs with the same display annunciation identify "hidden" automation behavior. The same combination of values/actions on the outputs with different display annunciation identify "ambiguous" automation behavior.

The SGA model also defines the complete set of situations perceived by the automation. The situations, by definition, are defined as a complete set of consistent combinations of conditions on the inputs to the automation. A subset of the set of inputs are the user entries on the automation user-interface. Selection of the same user input device that result in different behaviors identify modal behavior of user input devices.

Analyzing User-Interfaces Using the SGA model

User-interface control devices, such as the knobs, wheels, and buttons, are a subset of the inputs in the SGA model. These parameters typically play a significant role in the model as trigger conditions that directly cause changes in automation's behavior. By listing the control devices that invoke each of the automation's goals, the SGA model can be used to identify when the user-interface has:

- *more than one user input device that invokes the same automation behavior*
- *single user input device that is context dependent and will invoke different automation behaviors in different situations*

The user-interface of the automation provides feedback to the pilot on the behavior of the automation. By definition of the SGA model, each unique system behavior should be annunciated unambiguously to the operator. The SGA model of a device can be used to identify potential ambiguous displays when:

- *the same annunciation for different autopilot behaviors*
- *different annunciations for the same autopilot behavior.*

CASE STUDY: ANALYSIS OF MODERN AUTOPILOT MODE CONTROL PANEL

An SGA model of a modern autopilot was constructed and used to analyze the effectiveness of the MCP and FMA/PFD of the NASA Research Autopilot (Sherry, Feary, Polson, & Palmer, 1997). A sample of the autopilot goals for up-and-away operations (above 1500 ft) are listed in Table 1. A subset of the pilot MCP actions, or situations, that invoke these autopilot goals are listed for each autopilot goal along with the Flight Mode Annunciation (FMA) for speed and altitude.

AUTOPILOT GOAL	Situation (subset)	SPEED FMA	ALT FMA
CLIMB MAINTAIN MCP ALT	Dial MCP Altitude Knob/Pull MCP Altitude Knob	PITCH	CLB THRUST
DESCEND MAINTAIN MCP ALT	Dial MCP Altitude Knob/Pull MCP Altitude Knob	PITCH	IDLE
CLIMB MAINTAIN MCP ALT – ROC	Rotate MCP VS Wheel	THRUST	VS
DESCEND MAINTAIN MCP ALT – ROD	Rotate MCP VS Wheel	THRUST	VS
MAINTAIN CURRENT ALT	Push MCP Altitude Knob	THRUST	HOLD
CLIMB AWAY MCP ALT (2 SECS)	Rotate MCP VS Wheel	THRUST	VS
DESCEND AWAY MCP ALT (2 SECS	Rotate MCP VS Wheel	THRUST	VS
MAINTAIN MCP ALT	Aircraft is within +/- 60 ft and +/- 300 fpm of MCP Altitude	THRUST	HOLD

CLIMB MAINTAIN MCP ALT – CAP	Aircraft climbs descends within 0.03g Capture region to MCP Alt	THRUST	HOLD
DES MAINTAIN MCP ALT – CAP	Aircraft climbs descends within 0.03g Capture region to MCP Alt	THRSUT	HOLD
CLIMB AWAY MCP ALT	Rotate MCP VS Wheel	THRUST	VS
DESCEND AWAY MCP ALT	Rotate MCP VS Wheel	THRUST	VS
PROTECT SPEED ENVELOPE	Aircraft violates speed envelope	THRUST PITCH PITCH	VS CLB THRUST IDLE

Sample of Autopilot goals, the situations in which these goals are invoked, and the annunciation to the pilot for each goal.
Table 1.

It is evident from this table that several different autopilot goals are invoked by the same pilot action on the MCP. Also several different goals are annunciated by the same Speed and Altitude FMA.

Analysis of MCP Input Devices

The goals from the SGA model are mapped to the MCP input device knobs, wheels, and buttons (Sherry et. al., submitted –b). This mapping is summarized on Figure 2.

Different input devices that invoke the same goal are labeled in italics. For example, the autopilot can be instructed by the pilot to CLIMB MAINTAIN MCP ALT and DESCEND MAINTIAN MCP ALT by simply dialing up the MCP altitude (without pulling on the knob as is usually required) when the aircraft is not in a capture or holding the MCP Altitude. This is known as a “hot knob” operation.

Same input device that invokes different goals is identified by more than one set of goals listed for the input device. For example, rotating the MCP Vertical Speed Wheel will result in one of three classes of autopilot goals:

- CLIMB/DESCEND MAINTAIN MCP ALT – ROC/ROD
- CLIMB/DESCEND AWAY MCP ALT – ROC/ROD (2 SECS)
- CLIMB/DESCEND AWAY MCP ALT – ROC/ROD

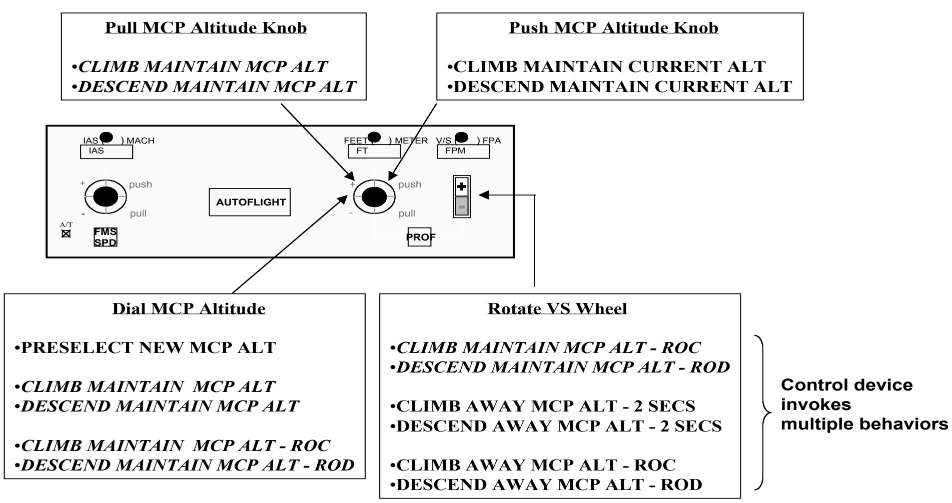
The goal that is invoked at any time is a function of the relative position of the MCP Altitude to the 0.03g capture region from the aircraft. As a result, unless the pilot is trained to recognize this situation, a command to the autopilot to DESCEND MAINTAIN MCP ALT – ROD may be interpreted by the autopilot as DESCEND AWAY MCP ALT – ROD.

Analysis of Annunciation

The set of behaviors of the autopilot, defined by the SGA model, where mapped to the display configurations of the FMA and PFD (Sherry, et. al., submitted –a).

Same Annunciations for Different Autopilot Behaviors: The labeling of the Speed || Altitude FMA does not provide unique labels for all of the possible autopilot goals. For example THRUST || VS appears 8 times in the table. It represents the following goals:

- CLIMB/DESCEND MAINTAIN MCP ALT – ROC/ROD
- CLIMB/DESCEND AWAY MCP ALT – ROC/ROD



MCP with autopilot labels for the goals that can be invoked by each MCP control device. Rotating the MCP Vertical Speed Wheel will result in one of three classes of autopilot behaviors depending on the situation.
Figure 2

(2 SECS)
- CLIMB/DESCENDAWAY MCP ALT – ROC/ROD
- PROTECT SPEED ENVELOPE
- MAINTAIN CURRENT ATTITUDE/SPEED

Distinguishing autopilot behaviors that CLIMB/DESCEND MAINTAIN MCP ALT, from behaviors that CLIMB/DESCEND AWAY MCP ALT are critical. Furthermore, distinguishing pilot invoked autopilot goals from those invoked autonomously by the autopilot, such as PROTECT SPEED ENVELOPE and MAINTAIN CURRENT ATTITUDE/SPEED, can eliminate automation surprises.

Different Annunciation for the Same Autopilot Behavior: The map of autopilot behaviors to display configurations on the FMA and PFD also identified different FMA for the same autopilot behavior. Autopilot goal PROTECT SPEED ENVELOPE has three combinations of Speed || Altitude FMA:

- PITCH || CLB THRUST
- PITCH || IDLE THRUST
- THRUST || VS.

Autonomous mode changes, by the autopilot, to protect the speed envelope are not explicitly identified in the FMA. Furthermore, the reason for these mode changes, especially when the transition is based on predicted aircraft state, is not provided to the pilot.

CONCLUSIONS & RECOMMENDATIONS

The formal SGA model of the modern autopilot provides the basis for evaluating the pilot/autopilot interaction. Non-unique configurations of the FMA/PFD result in ambiguous feedback to the operator of the intention of the autopilot. Multiple autopilot behaviors invoked by the same MCP input device in different situations prevent the pilot from conveying goals to the autopilot. Both of these characteristics contributed to the phenomena described in the representative example at the beginning of this paper.

Comparison with Other Methods

Analysis of the user-interface using an SGA model, derived from the actual software, emphasizes the behavior of the automation. The model explicitly defines the goals and behaviors that the automation can execute and uses these to identify two known causes of automation surprises.

The definition of goals is a useful supplement to the user-oriented task analysis methods, such as Cognitive Walkthrough (Lewis & Wharton, 1997), Operator-Function-Model (Callantine & Mitchell, (1999), and GOMS (John & Kieras, 1996). These methods are based on a hypothesized set of goals held by the operator that is difficult to validate. The SGA model enhances the definition of these goals by providing the finite set of goals that can be invoked by the automation.

The analysis of the SGA model has a lot in common with the theorem prover methods described by Palmer (1999) and Rushby (1999), and the finite state machine method of Degani & Heymann (submitted). These methods define formal models of the operators (as an automaton), the user-interface, and the automation. Execution of these models in a simulation uncovers potential problems. For example, Degani & Heymann demonstrate a problem where an internal state of the automation, critical to the situation awareness and resulting action sequence of the operator is not displayed on the user-interface. This results in the incorrect action sequence by the operator.

Palmer and Rushby use formal theorem-prover techniques to identify whether the model is capable of achieving specified states of the model. These models emphasize the behavior of the automation and are as useful as their analyst defined "model checking rules." Rules could be written to check for the two types of automation surprises defined in this paper.

Certification Criteria

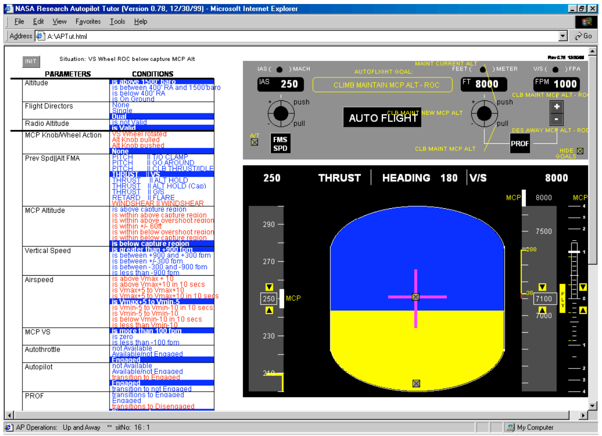
The analysis of the SGA model suggests criteria that can be used by manufacturers to demonstrate to the certification authorities, that unique labels are used for display of all autopilot behaviors. Specifically manufacturers will demonstrate that:

1) all unique behaviors of the automation, defined by the combinations of values/actions on the outputs, are annunciated to the pilot by unique, unambiguous labels on the cockpit displays (e.g. FMA and PFD)
2) all input devices to the automation (e.g. MCP) result in one, and only one, behavior of the automation (unless otherwise labeled).

Exceptions to these rules may require a waiver from the regulatory authority.

Mitigating These Automation Surprises

There are two ways in which autopilot behavior hidden by the FMA can be addressed. The simplest way to solve this type of automation surprise is to place knowledge in the world that makes the pilot task of inferring the autopilot goal intuitive: (1) display the autopilot goal on the MCP or on the PFD/FMA (Sherry & Polson, 1999; Feary et. al. 1998), or (2) train the operator. Training must explicitly define all of the autopilot behaviors and the set of cues that should be used to infer these goals. Figure 3 illustrates a web-based training device with goal-based "training scaffolding" that has been developed to mitigate this phenomenon(Sherry et. al., submitted –c).



Autopilot Tutor. Trains pilots on situations-goals-actions of autpilot. Training requires students to execute ATC instructions in a LOFT. Training scaffolding and reinforcing feedback is provided (Sherry, et. al., submitted –c).
Figure 3

There are three ways in which the autonomous modal behavior of the MCP control devices can be addressed: (1) place knowledge in the world that makes the pilot task of conveying a goal to the autopilot intuitive and direct with an MCP/FCU designed with input devices that result in one, and only one, behavior satisfies this design criterion, or (2) include dynamic labels for each MCP input device that reflect the autopilot goal that will be invoked when the input device is selected, or (3) train the operator.

Acknowledgements: This research was supported by the National Aeronautics and Space Administration (NASA) under contract NAS1-20219 to Honeywell Air Transport Systems (COTR: Ev Palmer) and cooperative agreement NCC 2-904 with the University of Colorado (COTR: Ev Palmer) Special thanks for several technical contributions and support: Mike Palmer (NASA – Langley), Steve Quarry, Dan McCrobie, Jim Martin, John Kilroy (Honeywell), Mike Padron, Rich Calvert, and Marty Alkin (FedEx), John Powers (UAL).

References:

Anderson, J. (1993) *Rules of the Mind.* L. Erlbaum and Associated: Hillsdale, New Jersey.

BASI (1998) Advanced Technology Aircraft Safety Survey Report. Report by Dept. of Transport and Regional Dev. Bureau of Air Safety Investigation. Available at http://www.basi.gov.au/sprog/advtek.htm

Card, S.K., T.P. Moran, & A. Newell (1983) *The psychology of human-computer interaction*. Lawrence Erlabaum: Hillsdale, N.J.

Callantine, T.J., & C.M. Mitchell (1999) *GT-CATS: Tracking Operator Activities in Complex Systems*. NASA/TM-1999-208788.

Degani, A,. & M. Heymann (submitted) Some formal aspects of human-automation interaction.

FAA (1996) *The interface between flightcrews and modern flight deck system*. FAA human factors team. Federal Aviation Administration. Available at http://www.faa.gov/avr/afs/interface.pdf

Hutchins, E. (1994) An integrated mode management interface for training. Training for automation workshop, NASA-Ames Research Center, Moffet Field, CA.

Javaux, D. (1998) The cognitive complexity of pilot mode interaction: A possible explanation of Sarter & Woods classical results. Proceedings *HCI-Aero, International Conference on Human-Computer Interaction in Aeronautics*, Montreal, Canada.

John, B. & D. Kieras (1996) Using GOMS for User-interface design and evaluation: Which technique ?" *ACM transactions on Computer-Human Interaction* V 3, N 4. Pgs 287-319.

Lewis, C. & C. Wharton (1997) Cognitive Walkthroughs. *Handbook of Human-Computer Interaction*. Eds. M. Helander, Landauer, & Prabhu. Elsevier Science B.V.(pages 717-732)

Monk, A. (1999) Modelling cyclic interaction. *Behavior & Information Technology,* Vol. 18, No. 2. Pgs 127-139.

Norman, D. A. (1988) *The Design of Everyday Things*. Doubleday: New York, N.Y.

NTSB (1999) Safety Recommendation. A-99-39 through 44

Palmer E. (1995) Oops, it did n't arm: A case study of two automation surprises." In proceedings *Eighth International Symposium on Aviation Psychology*, Colombus, OH, April, 1995. Pgs. 227-232.

Reason, J. (1987) The psychology of mistakes: A brief view of planning failures. In J. Rasmussen, K. Duncan, & J. leplat, eds, *New Technology and Human Error*. John Wiley: New Yor, N.Y. Pgs 45-52.

Rushby (1999) Using model checking to help discover mode confusions and other automation surprises. *Proceedings Human, Error, Safety, and System Development*. Liege, Belgium. (June, 1999)

Sarter, N.B., D.D. Woods, & C.E. Billings (1997) Automation surprises. In G. Salvendy, editor, *Handbook of Human Factors and Ergonomics*. John Wiley & Sons, 2nd edition.

Sherry, L. (1995) A formalism for the specification of operationally embedded reactive systems. In proceedings of the *International Council of System Engineering*, St. Louis, Missouri.

Sherry, L. M. Feary, P. Polson, & E. Palmer (1997) Analysis of the Behavior of a Modern Autopilot. Prepared for NASA under contract NAS1-20219. Honeywell Publication No. C69-5370-009.

Sherry, L.& P.Polson (1999) Shared models of flight management systems vertical guidance. In *The International Journal of Aviation Psychology – Special Issue: Aircraft Automation.* L. Erlbaum: N.Y.

Sherry, L., M. Feary, P. Polson, & E. Palmer (submitted-a) What's it doing now ? Taking the covers off a modern autopilot.

Sherry, L., M. Feary, P. Polson, & E. Palmer (submitted-b) Why won't it take that ? MCP control device invokes more than one autopilot behavior.

Sherry, L., M. Feary, P. Polson, & E. Palmer (submitted-c) Autopilot tutor: Building and maintaining autopilot skills.